**Fundamentos de Organización de Datos:**

**2011**

Hashing:

1. Cual de las siguientes definiciones corresponden al método de hash:

1. Técnica para generar una dirección base única para una clave dada.
2. Técnica que convierte la clave asociada a un registro de datos en un número aleatorio, que se utiliza para determinar donde se almacena el registro.
3. Técnica de almacenamiento y recuperación que usa una función para mapear registros en direcciones de almacenamiento en memoria secundaria.
4. **Todas las anteriores son aplicables.**

2. La técnica de hash

1. Entorpece la inserción y el borrado de elementos
2. La localización de un registro siempre debe utilizar una tabla adicional en memoria
3. **No es conveniente de aplicar sobre claves secundarias**
4. Requiere al menos de dos funciones de hash para el tratamiento de los desbordes.

3. Cuál de los siguientes conceptos corresponden con parámetros de la dispersión

1. Capacidad de almacenamiento de cada sector del archivo
2. Densidad de empaquetamiento
3. Método de tratamiento de desbordes
4. **Todos los anteriores**

4. La densidad de empaquetamiento se define como

1. El cociente entre cantidad de registro y espacio disponible en el archivo
2. El cociente entre la cantidad de registros y la cantidad de nodos del archivo
3. El cociente entre la cantidad de registros, y el producto entre la cantidad de nodos y el contenido posible de registros de cada nodo.
4. **Hay más de una respuesta correcta**

5. Una colisión se produce

1. **Cuando dos registros diferentes obtienen de la función de hash la misma dirección de disco**
2. Cuando dos registros iguales obtienen de la función de hash direcciones diferentes de disco
3. Cuando un registro no cabe en el lugar donde debe almacenarse de acuerdo al resultado de la función de hash.
4. Cuando dos registros diferentes obtienen de la función de hash direcciones diferentes de disco.

6. El hash asistido por tabla

1. **Utiliza un espacio estático de memoria**
2. Utiliza un espacio dinámico de memoria
3. No requiere de espacio adicional en memoria
4. Utiliza solo una función de hash para su desarrollo

7. La eficiencia promedio de búsqueda en un archivo a partir de estar organizado mediante política de Hashing:

1. Orden lineal
2. Orden logarítmico
3. **Orden constante**
4. No dispongo datos para contestar la pregunta

8. La eficiencia de búsqueda de un registro en un archivo organizado mediante hashing estático tiene:

1. Orden Lineal
2. **Algunas veces es uno**
3. Siempre es uno
4. Orden Logarítmico

9. Cuando la Densidad de empaquetamiento de un archivo tiende a uno

1. Es necesario redefinir el espacio disponible únicamente.
2. El archivo se completa y no es posible incorporar más elementos
3. Se debe cambiar la política de hash de estática a dinámica.
4. **Es necesario redefinir el espacio disponible y rehashear todo el archivo**

10. La técnica de hash extensible

1. Presenta una variante de hash que permite no solo ubicar rápidamente los registros, sino que además permite el acceso secuencial a los mismos.
2. Siempre inserta un registro con un y salo un acceso a disco
3. **Siempre se recupera un registro con un y solo un acceso a disco.**
4. En algunos casos recupera un registro con un y solo un acceso a disco.

**2017**

1. El borrado lógico en un archivo de datos:

1. Recupera inmediatamente el espacio borrado, disminuye el tamaño del archivo.
2. No se puede aplicar con registros de longitud variable.
3. Solo se aplica con registros de longitud fija.
4. Solo se aplica con registros de longitud variable.
5. **Ninguna de las opciones anteriores es correcta.**

2. Un algoritmo de corte de control:

1. Permite actualizar un archivo maestro con un archivo detalle.
2. Permite generar un archivo maestro a partir de varios archivos detalles.
3. **Permite presentar la información con un formato especial.**
4. Permite actualizar un archivo maestro con varios archivos detalles.

3. En un archivo con registros de longitud variable

1. **Puede utilizar “$” como delimitador de fin de registro.**
2. Cuando un registro se modifica utiliza siempre el mismo espacio.
3. Se puede utilizar cualquier lugar libre de un archivo para insertar un registro.
4. Siempre se utiliza la política de mejor ajuste para recuperar espacio.

4. Un árbol que no se encuentra balanceado

1. Puede ser un árbol binario
2. Puede ser un árbol multicamino
3. Puede ser un árbol binario paginado
4. No puede ser un árbol B+
5. No puede ser un árbol B\*
6. Todas las anteriores
7. Ninguna de las anteriores

5. Un árbol B de orden 200

1. En una hoja puede tener hasta 99 elementos
2. **En una hoja puede tener más de 99 elementos**
3. En una hoja puede tener menos de 99 elementos
4. La raíz siempre tiene hijos
5. La raíz tiene hijos si el árbol tiene más de 50 elementos

6. Cuando se inserta un elemento en un árbol Binario:

1. Siempre se debe generar un nuevo nodo.
2. Siempre es necesario acceder al nivel hoja.
3. Algunas veces puede llegar a reducir la altura del árbol.
4. Siempre aumenta la altura del árbol
5. Nunca aumenta la altura del árbol
6. Hay más de una respuesta correcta.

7. La densidad de empaquetamiento indica:

1. La cantidad claves en saturación.
2. La eficiencia del método de tratamiento de colisiones con saturación.
3. La cantidad de direcciones libres en dispersión dinámica.
4. Todas las opciones anteriores son correctas
5. Ninguna de las opciones anteriores es correcta.

8. Cuando una clave “x” y otra clave “y” generan, por función de dispersión, distinta dirección, entonces:

1. Una de los dos no será almacenada en el archivo.
2. **Se puede producir un desborde.**
3. Se produce una colisión.
4. “x” e “y” son sinónimos respecto a la función dispersión.
5. Hay más de una opción correcta.

9. La Doble Dispersión:

1. Evita que sucedan desbordes.
2. Evita que sucedan colisiones.
3. No es aplicable a la dispersión estática.
4. Siempre requiere más espacio físico que la Saturación Progresiva.
5. Ninguna de las opciones anteriores es correcta.

10. Una colisión:

1. Siempre produce saturación.
2. Solo ocurre cuando la dirección se encuentra vacía.
3. **Algunas veces produce saturación.**
4. Nunca puede producir saturación
5. Ninguna de las opciones anteriores es correcta.

22/06/2017

1. Cuando se realiza un alta en un árbol B:

1. Se puede realizar en un nodo interno.
2. Siempre produce overflow.
3. Puede llegar a necesitar de realizar una fusión de nodos.
4. **Siempre se llega hasta el nivel hoja.**
5. Ninguna respuesta es la correcta.

2. Cuales propiedades corresponden a un árbol B+ de prefijos simples

1. Cada nodo puede tener como máximo M descendientes, siendo M el orden del árbol
2. Un nodo que tiene x descendientes debe tener x-1 claves
3. Está siempre balanceado, sin importar los elementos que se inserten
4. Todas las propiedades anteriores

3. La eficiencia promedio de búsqueda en un árbol B tiene:

1. Orden lineal
2. **Orden logarítmico**
3. Orden constante
4. Ninguna respuesta es la correcta

4. En un árbol B de orden 50, cuando quedan 25 elementos en un nodo:

1. Se produce underflow y necesariamente debe concatenarse con un adyacente hermano
2. Se produce underflow y necesariamente debe redistribuirse con un adyacente hermano
3. Se produce underflow y la operación a realizar depende del estado de los nodos adyacentes hermanos.
4. **No se produce underflow**

5. Un índice primario es:

1. Una estructura de datos adicional que contiene el mismo volumen de información que el archivo original.
2. Una estructura de datos adicional que permite ordenar físicamente el archivo original.
3. **Una estructura de datos adicional que permite agilizar el acceso a la información del archivo.**
4. Una estructura de datos adicional que puede tener mayor volumen de información que el archivo original.

6. Un árbol binario:

1. Es una estructura de datos lineal, en la cual cada nodo puede tener a lo sumo dos hijos.
2. Es una estructura de datos no lineal, que siempre se encuentra balanceada.
3. Es una estructura de datos no lineal, donde cada nodo tiene dos hijos.
4. Es una estructura de datos lineal que se puede desbalancear.
5. **Es una estructura de datos no lineal que puede llegar a tener orden lineal de búsqueda.**
6. Ninguna de las opciones anteriores es correcta.

7. Un índice secundario es:

1. **Una estructura de datos adicional que permite asociar una o varias claves primarias con una clave secundaria.**
2. Una estructura de datos adicional que contiene el mismo volumen de información que el archivo original.
3. Una estructura de datos adicional que ordena físicamente (en memoria secundaria) el archivo original.
4. Una estructura de datos adicional que permite relacionar una clave secundaria con una sola clave primaria.
5. Todas las respuestas anteriores son correctas.

8. Cuando se realizan bajas en un árbol B:

1. Siempre se aplica redistribución.
2. Siempre se aplica fusión o concatenación.
3. Siempre se accede al nivel hoja.
4. La altura del árbol siempre se reduce.
5. **Algunas veces puede llegar a reducir la altura del árbol.**
6. Hay más de una respuesta correcta.

9. En un árbol B:

1. Cada nodo contiene X elementos y X-1 hijos.
2. En algunos casos la raíz puede tener un solo hijo.
3. Los nodos que contienen X elementos, contienen X+1 hijos.
4. Los nodos hojas pueden no estar al mismo nivel.
5. **Ninguna respuesta es la correcta.**
6. Hay más de una opción correcta.

10. En un árbol B+:

1. Para buscar un elemento siempre se llega al nivel hoja.
2. Los nodos hojas no deben estar enlazados entre sí.
3. Los nodos internos conforman un índice para llegar a un elemento buscado.
4. **Hay más de una opción correcta.**
5. Ninguna respuesta es la correcta.

**2020**

06/02/2020

1. Los archivos con registros de longitud variable

1. Ocupan menos espacio que los archivos con registros con longitud fija
2. Ocupan más espacio que los archivos con registros con longitud fija
3. Ocupan el mismo espacio que los archivos con registros con longitud fija
4. **Ninguna de las anteriores**

2. El proceso de alta de registro por ajuste optimo

1. Se puede realizar con registros de longitud fija
2. Se debe realizar con registros de longitud fija
3. Se puede realizar con registros de longitud variable
4. Se debe realizar con registros de longitud variable
5. **Ninguna de las anteriores**

3. Dado el NRR

1. Se puede acceder a un registro de un archivo fragmentado en sólo un acceso
2. Se puede acceder a un registro de un archivo no fragmentado en sólo un acceso
3. Se puede acceder a un registro de un archivo en sólo un acceso
4. Se puede lograr acceso directo a un archivo
5. **Todas las anteriores**
6. Algunas de las anteriores.

4. Sea un árbol B de orden 100, el nodo X tiene 49 elementos, si se borra un elemento de dicho nodo

1. Solo se borra el elemento
2. Se produce un underflow y se debe concatenar con el nodo X-1 o el X+1
3. Se produce un underflow y se debe redistribuir con el nodo X-1 o X+1
4. Todas las anteriores
5. Algunas de las anteriores
6. **Ninguna de las anteriores**

5. Un árbol B+ de prefijos simples de orden M

1. Es un árbol multicamino
2. Es un árbol balanceado
3. Es un árbol en que cada nodo (salvo la raíz) tiende a llenarse en al menos 2/3
4. Todas las anteriores
5. **Algunas de las anteriores**
6. Ninguna de las anteriores

6. Un árbol binario de orden 4

1. Puede desbalancearse
2. Puede balancearse
3. Si se emplean los algoritmos correctos puede quedar balanceado en altura
4. Si está desbalanceado no puede presentar una eficiencia de búsqueda de orden
5. Todas las anteriores
6. Algunas de las anteriores.
7. **No corresponde**

7. Si se quiere dispersar un archivo de 40.000 elementos

1. Se requiere un archivo de 40.000 cubetas
2. Se requiere un archivo de al menos 40.000 cubetas
3. Se requiere un archivo de menos de 40.000 cubetas
4. Todas las anteriores
5. Algunas de las anteriores
6. **Ninguna de las anteriores**

8. Si ocurrió saturación

1. No hubo colisión
2. Pudo haber ocurrido colisión
3. Hay más de 2 claves sinónimo
4. Todas las anteriores
5. Algunas de las anteriores
6. **Ninguna de las anteriores**

9. La densidad de empaquetamiento

1. Es un parámetro de eficiencia en cualquier tipo de hashing
2. **Es un parámetro de eficiencia sólo para un tipo de hashing**
3. Permite detectar si la cantidad de espacio libre en el archivo puede crecer
4. Permite detectar si la cantidad de elementos del archivo puede crecer
5. Algunas de las anteriores
6. Ninguna de las anteriores

10. Un archivo ordenado

1. **Puede desordenarse**
2. Conviene mantenerlo ordenado
3. No conviene mantenerlo ordenado
4. No puede desordenarse
5. Algunas de las anteriores
6. Ninguna de las anteriores

**2021**

16/07/2021

1. Un archivo con registro de longitud variable

1. Puede estar ordenado por algun criterio
2. Debe estar ordenado por algun criterio
3. Nunca puede ordenarse por algun criterio
4. Puede tener un carácter delimitador, por ejemplo &
5. **Algunas de las anteriores**
6. Ninguna de las anteriores

2. Una clave permite

1. Identificar un elemento particular dentro de un archivo
2. Reconocer un conjunto de elementos con igual valor
3. Ordenar lógicamente al archivo por los atributos que la componen
4. Todas las anteriores son válidas
5. **Algunas de las anteriores son válidas**
6. Ninguna de las anteriores es válida

3. Para que tenga sentido un algoritmo de corte de control

1. El archivo no necesita estar ordenado
2. El archivo puede estar ordenado
3. El archivo puede estar organizado por dispersión
4. El archivo debe estar organizado por dispersión
5. El archivo necesita al menos un índice asociado
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

4. Para que tenga sentido un algoritmo de corte de control

1. **El archivo debe estar ordenado**
2. El archivo puede estar ordenado
3. El archivo puede estar organizado por dispersión
4. El archivo debe estar organizado por dispersión
5. El archivo necesita al menos un índice asociado
6. Algunas de las anteriores
7. Ninguna de las anteriores

5. Un archivo en el cual se accede a un registro luego de acceder a su predecesor en algún orden

1. puede ser un archivo serie
2. puede ser un archivo secuencial
3. debe ser un archivo serie
4. debe ser un archivo secuencial
5. **algunas de las anteriores son correctas**
6. ninguna de las anteriores es correcta

6. El concepto de fragmentación interna en un archivo

1. Se puede dar solo en registros de longitud fija
2. Se puede dar solo en registros de longitud variable
3. Se puede dar solo en archivos ordenados de longitud variable
4. Se puede analizar solamente en archivos ordenados de longitud fija
5. Algunas de las anteriores
6. **Ninguna de las anteriores**

7. La técnica de primer ajuste

1. permite borrar elementos de un archivo que contiene registros de longitud fija
2. permite borrar elementos de un archivo que contiene registros de longitud variable
3. permite hacer baja logica
4. permite hacer baja física
5. Algunas de las respuestas anteriores son correctas
6. **Ninguna de las respuestas anteriores son validas**

8. El proceso de baja lógica

1. **Está diseñado para borrar un registro de un archivo**
2. Necesita que el archivo este ordenado
3. Necesita que el archivo este desordenado
4. Se aplica solamente a archivos con registros con longitud fija
5. Se aplica solamente a archivos con registros con longitud variable
6. Todas las anteriores
7. Alguna de las anteriores

9. El proceso de alta de registro con recuperación de espacio

1. Se debe realizar con registros de longitud variable
2. Se debe realizar con registros de longitud fija
3. Se realiza sólo con registros de longitud fija
4. Se realiza solo con registros de longitud variable
5. **Ninguna de las anteriores**

10. El proceso de baja lógica

1. Está diseñado para borrar un registro de un archivo
2. No necesita que el archivo este ordenado
3. Necesita que el archivo este desordenado
4. Se aplica solamente a archivos con registros con longitud fija
5. Se aplica solamente a archivos con registros con longitud variable
6. Todas las anteriores
7. **Algunas de las anteriores**

11. Un índice primario

1. Se puede generar a partir de una clave unívoca de un registro
2. Se puede generar a partir de una clave no unívoca de un registro
3. **Puede implementarse con una estructura de datos lineal**
4. Debe implementarse con una estructura eficiente no lineal
5. Algunas de las anteriores
6. Ninguna de las anteriores

12. Suponga que se genera un árbol binario para implantar un índice de un archivo. El índice es por la clave unívoca legajo que ocupa 10 bytes. Entonces, cada nodo del árbol ocupara:

1. 18 bytes
2. a lo sumo 18 bytes
3. **más de 21 bytes**
4. 10 bytes
5. ninguna de las anteriores

13. Cuando un arbol B tiende a llenarse

1. se debe procurar más espacio para el archivo que lo contiene, reacomodando todos los nodos
2. se debe procurar más espacio para el archivo que lo contiene, reacomodando el nodo padre y sus hermanos
3. se debe procurar más espacio para el archivo
4. todas las anteriores
5. algunas de las anteriores
6. **ninguna de las anteriores**

14. Cuáles de las siguientes propiedades no corresponde a un arbol b\* de orden M

1. Cada nodo del arbol puede contener máximo M descendientes y M-1 elementos
2. **La raiz no posee descendientes o posee [M/2] descendientes**
3. Todos los nodos terminales se encuentran al mismo nivel
4. Los nodos no terminales ni raiz tienen más de [M/2] y a lo sumo M descendientes
5. Un nodo con x descendientes contiene x-1 elementos

15. Un árbol que se encuentra balanceado

1. Puede ser un arbol binario
2. Puede ser un árbol multicamino
3. Puede ser un árbol B+
4. Puede ser un árbol B\*
5. **Todas las anteriores son correctas**
6. c y de son correctas
7. b, c y d son correctas
8. b y d son correctas
9. Ninguna de las anteriores es correcta

16. La eficiencia de busqueda de un arbol b

1. es de orden lineal
2. puede ser de orden lineal, bajo alguna circunstancia del arbol generado
3. **es de orden logarítmico**
4. es de orden constante (orden del arbol)
5. b y c son correctas
6. b, c y d son correctas
7. ninguna opción es correcta

17. Dado un archivo con registros de longitud fija. Suponga que tiene un índice correspondiente a una clave univoca. Suponga que se crea un arbol de orden 200 para almacenar ese índice univoca. Si archivo y el arbol generado tuvieran 300 elementos insertados entonces:

1. **si se creara un arbol b tendría la misma altura que crear un arbol b\***
2. si se creara un arbol b tendría mayor altura que un arbol b\*
3. si se creara un arbol b tendría menor altura que un arbol b\*
4. no tengo datos suficientes para contestar la pregunta

18. Un árbol B+ de prefijos simples

1. A veces no tiene prefijos
2. Puede no tener prefijos simples
3. Si tiene prefijos simples, las hojas están enlazadas
4. Sino tiene prefijos simples, las hojas están enlazadas
5. **Todas las anteriores**
6. Algunas de las anteriores
7. Ninguna de las anteriores

19. Un Arbol B+

1. Al realizar una inserción, puede tener nodos hojas con underflow
2. Al realizar una baja, puede tener nodos hojas con overflow
3. Al realizar una inserción, puede requerirse concatenación
4. Al realizarse una baja, puede requerirse división
5. Al realizarse un alta puede requerirse redistribución
6. Todas las anteriores
7. Algunas de las anteriores
8. **Ninguna de las anteriores**

20. Un arbol b+ de prefijos simples de orden M

1. se puede aplicar a una clave única, cuyo atributo es un string
2. se puede aplicar a una clave secundaria, cuyo atributo es un string
3. se puede aplicar a una clave primaria, cuyo atributo es un entero
4. se puede aplicar solamente a una clave unívoca, cuyo atributo es string
5. todas las anteriores son correctas
6. **a y b son las opciones correctas**
7. a, c y d son las opciones correctas
8. a, b y d son las opciones correctas
9. ninguna de las opciones que tengo es correcta

21. Cuáles de los siguientes parámetros no corresponde a hashing

1. Función de hash
2. Tamaño del nodo (capacidad para almacenar registros)
3. **Densidad de busqueda**
4. Forma de tratar los desbordes
5. Todas las anteriores corresponden

22. Cuál de los siguientes parámetros afecta la eficiencia de la dispersión

1. Cantidad de elementos del archivo
2. Cantidad de espacio para almacenar el archivo
3. Densidad de Empaquetamiento
4. Función de Dispersión
5. Algoritmos para el tratamiento de registros sinónimos
6. Todas las anteriores
7. **Algunas de las anteriores**
8. Ninguna de las anteriores

23. Un archivo tratado con hash estático, que tiene una densidad de empaquetamiento del 10%

1. tiene mucha fragmentación interna
2. tiene mucha fragmentación externa
3. presenta un nivel de colisiones bajo
4. presenta un nivel de overflow bajo
5. Todas las anteriores son correctas
6. las opciones b y c son correctas
7. las opciones a y c son correctas
8. **las opciones a, c y d son correctas**
9. las opciones b, c y d son correctas
10. ninguna opción es correcta

24. La función de hashing

1. Afecta a la densidad de empaquetamiento
2. Puede afectar a la densidad de empaquetamiento
3. Afecta la densidad de empaquetamiento sólo en casos especiales
4. Afecta la densidad de empaquetamiento en la mayoría de los casos
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

25. La técnica de Área de desborde por separado:

1. **Utiliza un área de memoria separada para las claves en overflow.**
2. Reduce la densidad de empaquetamiento.
3. Utiliza una segunda función de dispersión siempre que se desee almacenar un registro en un archivo.
4. Solo se aplica a la dispersión extensible.
5. Ayuda a predecir la cantidad de claves en overflow.
6. Hay más de una opción correcta.
7. Ninguna de las opciones anteriores es correcta

26. El método de área de desborde separada

1. **Reubica los registros en overflow**
2. Utiliza una segunda funcion de hash en caso de ser necesaria
3. Puede generar areas de overflow dentro del archivo
4. Todas las anteriores son validas
5. Algunas de las anteriores
6. Ninguna de las anteriores.

27. Si se tiene una política de hash con espacio de direccionamiento dinámico

1. la densidad de empaquetamiento puede ser mayor que 1
2. cuando la densidad de empaquetamiento supera el 75% se debe reacomodar al archivo
3. cuando la densidad de empaquetamiento supera el 75% se debe utilizar más espacio para nodos
4. cuando la densidad de empaquetamiento supera el 75% debe activarse una política de tratamiento de overflow, como por ejemplo área de desborde separado
5. Algunas de las anteriores son validas
6. **ninguna de las anteriores es valida**

28. A partir de un archivo dispersado con hashing extensible

1. Siempre es posible agregar elementos al archivo
2. Algunas veces no es posible agregar elementos al archivo
3. Se puede utilizar Saturación Progresiva Encadenada para tratar registros en saturación
4. No se puede utilizar Dispersión Doble para tratar registros en saturación
5. **Algunas de las anteriores**
6. Ninguna de las anteriores

29. La técnica de hashing extensible

1. No utiliza una función de hash porque debe cambiar dinámicamente
2. No utiliza una función de hash porque encuentra los registros siempre en un acceso
3. **Utiliza una función de hash, pero esta función no devuelve la dirección donde guardar el registro**
4. Utiliza área de desborde por separado para los registros en overflow
5. Todas las anteriores son validas
6. Las respuestas a y c son correctas
7. Las respuestas b y c son correctas
8. Las respuestas a, c y d son correctas
9. Ninguna es correcta

30. Cuál de las siguientes consignas no define hash dinámico

1. recupera los registros en un acceso a disco
2. **No puede haber estructuras adicionales**
3. Se organiza todo el archivo de datos
4. Solo puede organizarse por un UNICO criterio, la clave primaria
5. Todas las anteriores pertenecen a hash dinámico

------------------------------------------------------------------------------------------------------------------------------------------------

1. Un archivo con registro de longitud fija

1. Puede tener un delimitador de fin de registro
2. Debe tener un delimitador de fin de registro
3. Puede tener registros del mismo tamaño
4. Puede tener registros con distinto tamaño
5. Algunas de las anteriores
6. **Ninguna de las anteriores**

2. Para realizar un algoritmo de actualización maestro-detalle

1. **Se requiere al menos 2 archivos.**
2. Se requiere que los archivos estén ordenados
3. Se requiere que los archivos estén desordenados
4. Se requiere que al menos uno de los archivos esté ordenado
5. Se requiere que los archivos tengan la misma estructura
6. Algunas de las anteriores
7. Ninguna de las anteriores

3. El acceso promedio para recuperar un dato en un archivo desordenado:

1. **Tiene orden lineal.**
2. Tiene orden logarítmico.
3. Tiene orden constante (uno).
4. Las opciones a y b son correctas.
5. Ninguna opción es correcta.

4. Un algoritmo de corte de control

1. Se puede aplicar sobre un archivo con registros de longitud fija
2. Se puede aplicar sobre un archivo con registros de longitud variable
3. Se aplica sobre un archivo con registros de longitud fija
4. Se aplica sobre un archivo con registros de longitud variable
5. Todas las anteriores son correctas
6. **Algunas de las anteriores son correctas**
7. Ninguna de las anteriores son correctas

5. El acceso promedio para recuperar un dato en un archivo desordenado:

1. Puede tener orden lineal.
2. Puede tener orden logarítmico.
3. Tiene orden constante (uno).
4. Las opciones a y b son correctas.
5. **Ninguna opción es correcta.**

6. El proceso de merge de archivos

1. Requiere que todos los archivos estén ordenados
2. Requiere que todos los archivos estén ordenados por el mismo criterio
3. **Puede realizarse con los archivos ordenados**
4. No puede realizarse sin los archivos ordenados
5. Algunas de las anteriores
6. Ninguna de las anteriores

7. La técnica de mejor ajuste

1. asigna un registro nuevo (alta) en una la posición que quepa de tamaño menor
2. asigna un registro nuevo (alta) en una la posición que quepa de tamaño mayor
3. asigna el registro al final del archivo si no hay lugar en posiciones intermedias
4. Todas las anteriores son correctas
5. **Algunas de las respuestas anteriores son correctas**
6. Ninguna de las respuestas anteriores son validas

8. La técnica de primer ajuste

1. se aplica a archivos con registro de longitud fija
2. **se aplica a archivos con registro de longitud variable**
3. se combina con la técnica de baja física de datos
4. Todas las anteriores son correctas
5. Algunas de las respuestas anteriores son correctas
6. Ninguna de las respuestas anteriores son validas

9. La técnica de mejor ajuste

1. permite borrar elementos de un archivo que contiene registros de longitud fija
2. permite borrar elementos de un archivo que contiene registros de longitud variable
3. permite hacer baja logica
4. permite hacer baja física
5. Algunas de las respuestas anteriores son correctas
6. **Ninguna de las respuestas anteriores son validas**

10. El proceso de alta de registro por peor Ajuste

1. Se puede realizar con registros de longitud fija
2. Se debe realizar con registros de longitud fija
3. Se puede realizar con registros de longitud variable
4. **Se debe realizar con registros de longitud variable**
5. Ninguna de las anteriores

11. En un índice secundario

1. Encontrar un registro es a veces más lento que sobre un índice primario
2. Encontrar un registro es igual de rápido que sobre un índice primario
3. Encontrar un registro puede ser más rápido que sobre un índice primario
4. si la clave a buscar no se repitiera puede ser igual de rápida su búsqueda que en un índice unívoco
5. Algunas de las anteriores son validas
6. **ninguna de las anteriores es valida**

12. Un árbol binario

1. puede tener eficiencia de búsqueda logarítmica
2. puede estar balanceado si tiene 127 elementos
3. puede tener eficiencia de búsqueda lineal
4. **todas las anteriores son correctas**
5. algunas de las anteriores son validas
6. ninguna de las anteriores es valida

13. Dado un arbol b\* de orden 100

1. **el nodo raíz pude tener solamente 3 hijos en algun momento de su construcción**
2. un nodo terminal puede producir un underflow, si al borrar un elemento quedan 48 en el nodo
3. puede ser que un nodo terminal con 50 elementos, tenga 51 hijos
4. todas las respuestas anteriores son correctas
5. las opciones a y b son correctas
6. las opciones b y c son correctas
7. las opciones a y c son correctas
8. ninguna opción anterior es correcta

14. Una inserción en un nodo cualquiera (terminal) de arbol b

1. puede generar overflow
2. puede generar división de un solo nodo
3. pueden dividirse tres nodos
4. **todas las anteriores son correctas**
5. las dos primeras son correctas
6. la segunda y tercera son correctas
7. la primera y la tercera son correctas

15. Dado un arbol b de prefijos simples de orden 100

1. un nodo entra en underflow si se borra un elemento y solo quedan 48
2. un nodo no terminal ni raiz tendrá al menos [M/2] hijos
3. puede ser que la raiz tenga solamente dos hijos en algun momento
4. **todas las anteriores son correctas**
5. algunas de las anteriores son correcta
6. ninguna de las anteriores son correctas

16. Un arbol b

1. puede ser un arbol AVL
2. puede guardarse en memoria RAM
3. puede implementar una clave no univoca
4. puede tener acceso secuencial eficiente y rápido
5. Todas las anteriores son correctas
6. **Algunas de las anteriores son correctas**
7. ninguna de las anteriores son correctas

17. Cuáles de las siguientes propiedades NO corresponde a un arbol b+ de orden M

1. cada nodo del arbol puede contener como máximo M descendientes y M-1 elementos
2. la raiz no posee descendientes o tiene al menos dos
3. un nodo con x descendientes tiene x-1 elementos
4. los nodos terminales tienen como máximo M-1 elementos
5. los nodos no terminales pueden contener como minimo [2M/3] descendientes
6. los nodos no terminales pueden contener como minimo [M/2] descendientes
7. **todas las respuestas anteriores corresponden a propiedades de árboles b+**

18. Cuando un arbol B+ de prefijos simples tiende a llenarse

1. se debe procurar más espacio para el archivo que lo contiene, reacomodando todos los nodos
2. se debe procurar más espacio para el archivo que lo contiene, reacomodando el nodo padre y sus hermanos
3. se debe procurar más espacio para el archivo
4. todas las anteriores
5. algunas de las anteriores
6. **ninguna de las anteriores**

19. El hash con espacio de direccionamiento estático

1. Puede tener densidad de empaquetamiento menor que uno
2. Puede tener un tratamiento de desbordes
3. Puede tener una función aleatoria y uniforme
4. Todas de las anteriores
5. Las opciones a y b son correctas
6. las opciones a y c son correctas
7. las opciones b y c son correctas
8. **ninguna de las anteriores son correctas**

20. Si la densidad de empaquetamiento tiene a 1 (o el 100%):

1. Es conveniente utilizar dispersión doble para el tratamiento de overflow.
2. Se debe cambiar la función de dispersión.
3. **Debe aumentarse el tamaño del archivo.**
4. Debe aumentarse el tamaño de los nodos
5. Todas las anteriores son validas
6. Las opciones b y c son correctas
7. Las opciones b y d son correctas
8. Las opciones c y d son correctas
9. Las opciones b, c y d son correctas
10. Ninguna de las opciones anteriores es correcta.

21. Cuál de los siguientes parámetros afecta la eficiencia de la dispersión

1. Cantidad de elementos del archivo
2. Cantidad de espacio para almacenar el archivo
3. Densidad de Empaquetamiento
4. Función de Dispersión
5. Algoritmos para el tratamiento de registros en saturación
6. **Todas las anteriores**
7. Algunas de las anteriores
8. Ninguna de las anteriores

22. Cuál de los siguientes métodos sirve para el tratamiento de colisiones en hash estático

1. área de desborde por separado
2. saturación progresiva
3. saturación progresiva encadenada
4. doble dispersión
5. Todas las anteriores
6. **ninguna de las anteriores son validas**

23. El método de área de desborde por separado

1. Utiliza una segunda función de hash para ubicar los registros en saturación de un archivo
2. direcciona el overflow de un nodo a otro nodo diferente
3. Evita generar zonas contiguas de nodos en overflow
4. todas las anteriores son validas
5. Las opciones a y b son verdaderas
6. Las opciones a y c son verdaderas
7. **Las opciones b y c son verdaderas**
8. Ninguna de las anteriores

24. La eficiencia de búsqueda de un registro en un archivo organizado mediante dispersión estática:

1. Es de Orden lineal
2. **tiende a uno**
3. Siempre es uno
4. Es de Orden Logarítmico
5. Algunas de las anteriores son correctas
6. Ninguno de las opciones anteriores es correcta

------------------------------------------------------------------------------------------------------------------------------------------------

1. La operación RESET()

1. Se realiza luego de la operación REWRITE()
2. Se realiza previo a la operación ASSIGN()
3. Abre un archivo para leer o escribir
4. Abre un archivo sólo para escribir
5. **Abre un archivo**

2. La operación REWRITE()

1. Se realiza luego de la operación RESET()
2. Se realiza previo a la operación ASSIGN()
3. Abre un archivo para leer o escribir
4. Abre un archivo sólo para leer
5. **Abre un archivo**

3. Un archivo en el cual se accede a un registro luego de acceder a su predecesor en orden físico

1. puede ser un archivo serie
2. puede ser un archivo secuencial
3. **debe ser un archivo serie**
4. debe ser un archivo secuencial
5. algunas de las anteriores son correctas
6. ninguna de las anteriores es correcta

4. La técnica de primer ajuste

1. asigna un registro nuevo (alta) en una la posición que quepa de tamaño menor
2. asigna un registro nuevo (alta) en una la posición que quepa de tamaño mayor
3. **asigna el registro al final del archivo si no hay lugar en posiciones intermedias**
4. Todas las anteriores son correctas
5. Algunas de las respuestas anteriores son correctas
6. Ninguna de las respuestas anteriores son validas

5. La técnica de altas reutilizando espacio borrado, conocida como mejor ajuste, en archivos de longitud fija:

1. asigna al registro en el primer espacio que encuentra donde quepa
2. asigna el registro en el espacio donde quepa, de tamaño menor
3. asigna el registro en el espacio donde quepa, de tamaño mayor
4. algunas de las anteriores
5. **Ninguna de las anteriores**

6. Cuando se inserta un elemento en un árbol binario

1. siempre se crea un nuevo nodo
2. siempre es necesario acceder al nivel hoja
3. puede siempre insertarse a derecha del padre
4. Puede aumentar la altura del árbol
5. **Todas las anteriores son validas**
6. algunas de las anteriores son validas
7. ninguna de las anteriores son validas

7. Un arbol balanceado de orden 200

1. se desbalancea cuando la raiz es el único nodo del arbol y esta produce overflow en una inserción.
2. puede contener un nodo con 99 hijos y 99 claves
3. dos nodos adyacentes hermanos, de diferente padre, pueden estar completos, es decir con 199 elementos
4. todos los nodos terminales están a la misma distancia de todos los nodos raiz
5. todas las anteriores son correctas
6. algunas de las anteriores son correctas
7. **ninguna de las anteriores son correctas**

8. Un arbol b de orden 130

1. puede tener la raiz con solamente dos hijos
2. tienen todos los nodos terminales en igual nivel
3. puede tener todos sus nodos ocupados en al menos 2/3 de su capacidad
4. **Todas las anteriores son correctas**
5. las opciones a y b son correctas
6. las opciones b y c son correctas
7. las opciones a y c son correctas
8. ninguna opción es correcta

9. Suponga que sobre un nodo de un árbol B\* se produce overflow. En dicho caso se puede

1. Aplicar saturación progresiva encadenada
2. Aplicar doble dispersión
3. Aplicar un área de desborde separada para el nodo
4. Redistribución
5. División
6. Todas las anteriores
7. **Algunas de las anteriores**
8. Ninguna de las anteriores

10. Sea un problema donde un índice correspondiente a una clave unívoca se implementa como un arbol balanceado.

1. el archivo de datos debe ser de registros de longitud fija
2. **el archivo de datos puede ser de registros de longitud variable**
3. el archivo de datos no puede admitir bajas con recuperación de espacio
4. las opciones a y b son correctas
5. las opciones a y c son correctas
6. las opciones b y c son correctas
7. ninguna opción es correcta

11. Un Arbol B+

1. Siempre tiene más claves que un árbol B, para el mismo archivo de datos
2. Siempre tiene más claves que un árbol B\*, para el mismo archivo de datos
3. Siempre es más alto que un árbol B
4. Siempre es más alto que un árbol B\*
5. Todas de las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

12. Suponga que el nodo terminal de un arbol B+ de prefijos simples de orden 7 tiene las claves GONZALEZ, GOÑEZ, GOODMAN, GOPLANI, GORBA y en dicho nodo se inserta una clave nueva GUTIERREZ, entonces:

1. se produce overflow, y luego de dividir, al padre del nodo se sube la clave GOP
2. se produce overflow, y luego de dividir, al padre del nodo se sube la clave GOO
3. se produce overflow, y luego de dividir se sube la clave GO
4. Todas las anteriores pueden ser correctas, depende el algoritmo utilizado
5. a y b pueden se correctas, dependen del algoritmo utilizado
6. a y c pueden ser correctas, dependen del algoritmo utilizado
7. b y c pueden ser correctas, dependen del algoritmo utilizado
8. **ninguna de las anteriores es correcta**

13. La eficiencia de búsqueda de una clave en un arbol b+ es

1. de orden lineal
2. de orden logarítmico similar a un arbol b
3. de orden logarítmico, similar a un arbol b\*
4. de orden fijo, dado que los elementos de los nodos terminales estan linkeados juntos
5. las respuestas a, b y c son correctas
6. las respuestas b, c y d son correctas
7. **las respuestas b y c son correctas**
8. ninguna de las respuestas anteriores es correcta

14. El método de doble dispersión, para el tratamiento de Colisiones

1. Utiliza una segunda función de hash para ubicar a todos los registros del archivo
2. Utiliza una segunda función de hash para ubicar algunos registros del archivo
3. Evita generar zonas contiguas de nodos en overflow
4. todas las anteriores son validas
5. Las opciones a y b son verdaderas
6. Las opciones a y c son verdaderas
7. Las opciones b y c son verdaderas
8. **Ninguna de las anteriores**

15. El método de Dispersión Doble

1. Afecta a la densidad de empaquetamiento
2. Puede afectar a la densidad de empaquetamiento
3. Afecta la densidad de empaquetamiento sólo en casos especiales
4. Afecta la densidad de empaquetamiento en la mayoría de los casos
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

16. Con 10000 direcciones con capacidad para 4 registros cada una y 30000 claves para dispersar, entonces

1. La densidad de empaquetamiento es mayor que uno
2. **La densidad de empaquetamiento es superior o igual al 75%**
3. La densidad de empaquetamiento es inferior al 75% pero mayor o igual al 50%
4. La densidad de empaquetamiento es inferior al 50% pero mayor o igual al 25%
5. No se puede calcular

17. El método de área de desborde por separado

1. Utiliza una segunda funcion de hash para ubicar los registros en saturación de un archivo
2. Ubica los registros lo más próximo posible a su dirección base
3. **Evita generar zonas contiguas de nodos en overflow**
4. todas las anteriores son validas
5. Las opciones a y b son verdaderas
6. Las opciones a y c son verdaderas
7. Las opciones b y c son verdaderas
8. Ninguna de las anteriores

18. El hashing extensible

1. Afecta a la densidad de empaquetamiento
2. Puede afectar a la densidad de empaquetamiento
3. Afecta la densidad de empaquetamiento sólo en casos especiales
4. Afecta la densidad de empaquetamiento en la mayoría de los casos
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

------------------------------------------------------------------------------------------------------------------------------------------------

1. El algoritmo de actualización maestro – detalle

1. Solo se puede implementar si los archivos (maestro y detalles) estan ordenados
2. Solo se puede implementar si el archivo maestro esta ordenado
3. Solo se puede implementar si los archivos detalles estan ordenados
4. Solo se puede implementar si los archivos (maestro y detalles) estan desordenados
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

2. Fragmentación interna es:

1. **aquella que se produce cuando un elemento de dato se le asigna mayor espacio del necesario**
2. el espacio disponible entre dos registros, disponible pero no utilizado
3. el espacio disponible entre dos registros, disponible y utilizado
4. algunas de las anteriores definen fragmentación interna
5. ninguna de las anteriores es correcta

3. Un archivo directo

1. debe contener registros de longitud fija
2. debe contener registros de longitud variable
3. puede contener registros de longitud fija
4. puede contener registros de longitud variable
5. permite que a y b sean correctas
6. permite que a y c sean correctas
7. **permite que c y d sean correctas**
8. permite que b y d sean correctas

4. Un archivo directo

1. permite acceso secuencial únicamente
2. permite acceso secuencia indizado
3. **permite acceso directo**
4. respeta a, b y c
5. no tengo datos suficientes para contestar

5. El concepto de fragmentación externa en un archivo

1. Se da solo en registros de longitud fija
2. **Se da solo en registros de longitud variable**
3. Se da en archivos ordenados de longitud variable
4. Se debe analizar solamente en archivos ordenados de longitud fija
5. Algunas de las anteriores
6. Ninguna de las anteriores

6. En el caso de realizar un alta de un registro

1. se reaprovecha algún espacio de registro dado de baja previamente
2. **se puede reaprovechar algún espacio de registro dado de baja previamente**
3. se debe reaprovechar si los registros son de longitud fija
4. se debe reaprovechar si los registros son de longitud variable
5. algunas de las anteriores son correctas
6. ninguna de las anteriores son correctas

7. Un índice secundario tiene eficiencia de búsqueda

1. lineal
2. logarítmica
3. contante y es 1
4. constante y pude tender a 1
5. Algunas de las anteriores son validas
6. **ninguna de las anteriores son validas**

8. Un árbol avl

1. **tiene eficiencia de búsqueda logarítmica**
2. puede tener eficiencia de búsqueda logarítmica
3. puede tener eficiencia de búsqueda lineal
4. Las opciones a y b son correctas
5. Las opciones b y c son correctas
6. Las opciones a y c son correctas
7. Ninguna opción es correcta

9. Sea un archivo de alumnos, que maneja registros de longitud fija y que utiliza la técnica de primer ajuste para recuperar espacio en el caso de borrado. Si se implementa un índice por clave unívoca legajo

1. se puede generar un arbol b para implantar ese índice
2. se puede generar un arbol b\* para implantar ese índice
3. se puede generar un arbol b+ para implantar ese índice
4. se puede implantar un arbol b+ de prefijos simples para implantar ese índice
5. Todas las opciones anteriores son correctas
6. algunas de las opciones anteriores son correctas
7. **la pregunta es inconsistente no hay respuestas correctas**

10. Dado un arbol b de orden 100

1. el nodo raíz puede tener solamente 3 hijos en algun momento de su construcción
2. El proceso de borrar un elemento del nodo raiz, puede producir un underflow en un nodo terminal
3. puede ser que un nodo terminal con 50 elementos, tenga 51 hijos
4. todas las respuestas anteriores son correctas
5. **las opciones a y b son correctas**
6. las opciones b y c son correctas
7. las opciones a y c son correctas
8. ninguna opción anterior es correcta

11. Dado un archivo con registros de longitud fija. Suponga que tiene un índice correspondiente a una clave univoca. Suponga que se crea un arbol de orden 200 para almacenar ese índice univoca. Si archivo y el arbol generado tuvieran 300 elementos insertados entonces:

1. un arbol b tiene la misma cantidad de nodos que un arbol b\* creado para la misma finalidad
2. un arbol b tiene más cantidad de nodos que un arbol b\* creado para la misma finalidad
3. un arbol b tiene menos cantidad de nodos que un arbol b\* creado para la misma finalidad
4. **no tengo datos suficientes para contestar la pregunta**

12. Un arbol b\*

1. todos los nodos menos la raiz seguro estan llenos a 2/3 de su capacidad en todo momento
2. **los nodos terminales pueden, en situaciones especiales, tener menos de 2/3 de su capacidad ocupada**
3. el nodo adyacente hermano de uno que entra en overflow siempre se puede usar para redistribuir, si no estuviera completo
4. todas las anteriores son correctas
5. las opciones a y b son correctas
6. las opciones b y c son correctas
7. las opciones a y c son correctas
8. ninguna opción es correcta

13. Un arbol b es más eficiente que un arbol b+

1. porque tiene un algoritmo de inserción más eficiente
2. porque tiene un algoritmo de borrado más eficiente
3. porque tiene un algoritmo de busqueda más eficiente
4. no, un arbol b no es más eficiente que un arbol b+
5. por todas las anteriores
6. por algunas de las anteriores
7. **por ninguna de las anteriores**

14. Cuando una clave “x” y otra clave “y” generan, por función de dispersión, una misma dirección, entonces:

1. Una de las dos no será almacenada en el archivo.
2. Se produce un desborde.
3. **Se produce una colisión.**
4. Hay más de una opción correcta.

15. Para tener un bajo porcentaje de densidad de empaquetamiento

1. **se deben tener disponibles muchos más nodos de los necesarios**
2. se debe utilizar una función de hash que genere pocas colisiones
3. se deben tener nodos con capacidad de almacenar más de 100.000 registros
4. se debe utilizar hash con espacio de direccionamiento dinámico
5. todas las anteriores sirven para para bajar el porcentaje de la densidad de empaquetamiento
6. ninguna respuesta es correcta

16. Cuando se produce un overflow

1. Se puede aplicar el método de doble dispersión
2. Se puede aplicar el método de área separada
3. Se puede crear un nodo nuevo
4. Se puede aplicar el método de saturación progresiva
5. **Todas las anteriores pueden darse**
6. Algunas de las anteriores son validas
7. ninguna de las anteriores son validas

17. La dispersión dinamica, denominada hash extensible:

1. **Siempre requiere el uso de una estructura auxiliar.**
2. Guarda los registros de forma ordenada por algún criterio
3. Necesita de dos funciones de dispersión.
4. Siempre requiere variar el tamaño del espacio de direcciones.
5. Ninguna de las opciones anteriores es correcta

18. La dispersión dinamica, denominada hash extensible:

1. Siempre requiere el uso de una estructura auxiliar.
2. Guarda los registros de forma ordenada por algún criterio
3. Necesita de dos funciones de dispersión.
4. Varía el tamaño del espacio de direcciones disponible, sin afectar a la función de has
5. las opciones a y b son correctas
6. **las opciones a y d son correctas**
7. Ninguna de las opciones anteriores es correcta

------------------------------------------------------------------------------------------------------------------------------------------------

1. Una clave unívoca permite

1. identificar un elemento particular dentro de un archivo
2. reconocer un conjunto de elementos con igual valor
3. ordenar lógicamente al archivo por los atributos que la componen
4. todas las anteriores son validas
5. **algunas de las anteriores son validas**
6. ninguna de las anteriores es valida

2. El concepto de fragmentación interna en un archivo

1. Se puede dar en registros de longitud fija
2. Se puede dar en registros de longitud variable
3. Se puede dar en archivos ordenados de longitud variable
4. Se puede analizar en archivos ordenados de longitud fija
5. **Todas las anteriores**
6. Algunas de las anteriores son validas
7. Ninguna de las anteriores

3. Cuando se utiliza un algoritmo de merge

1. Es necesario que el archivo maestro sea de longitud fija
2. Es necesario que el archivo detalle sea de longitud fija
3. Necesariamente el archivo maestro quedará ordenado
4. Es necesario que el archivo detalle esté ordenado
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

4. En el caso de realizar un alta de un registro

1. se puede reaprovechar algún espacio de registro dado de baja previamente
2. se debe reaprovechar algún espacio de registro dado de baja previamente
3. se puede reaprovechar si los registros son de longitud fija
4. se puede reaprovechar si los registros son de longitud variable
5. **algunas de las anteriores son correctas**
6. ninguna de las anteriores son correctas

5. La eliminación lógica de elementos de un archivo

1. es menos eficiente en términos de espacio que la física
2. **es menos eficiente en términos de tiempo de respuesta del algoritmo que la física**
3. es más eficiente en términos de espacio y de tiempo de respuesta del algoritmo que la física
4. Algunas de las anteriores
5. ninguna de las anteriores

6. ¿Cuál de las siguientes definiciones no se corresponde con un índice?

1. Es una estructura de datos adicional que permite agilizar el acceso a la información almacenada en un archivo
2. es una estructura de datos adicional que permite agilizar el recorrido ordenado de un archivo
3. es una estructura adicional implementada con un arbol balanceado que permite agilizar el acceso a la información almacenada en un archivo
4. es una estructura auxiliar que permite ordenar físicamente un archivo, asi se pueden hacer búsquedas más eficientes
5. Todas las anteriores definiciones corresponden a un índice
6. **hay más de una definición que no corresponde a índice**
7. ninguna de las definiciones anteriores corresponden a índices

7. Un árbol binario

1. **permite implementar un índice secundario de un archivo**
2. es la mejor solución para implementar un índice a partir de la clave primaria de un archivo
3. es la única solución para implementar el índice correspondiente a una clave unívoca
4. cada elemento de arbol tiene un hijo izquierdo y un hijo derecha
5. Todas las anteriores son validas
6. Algunas de las anteriores son validas
7. Ninguna de las anteriores son validas

8. Dado un arbol b. Suponga que se tienen tres nodos terminales A, B y C. A adyacente hermano izquierda de B, C adyacente hermano derecha de B. Suponga que el nodo A esta completo, y que el nodo C tiene lugar disponible. Se inserta un elemento en el nodo B

1. debe redistribuir con el nodo C si la politica que se usa es la de derecha o izquierda
2. debe dividirse con el nodo A si la politica que se usa es la de izquierda
3. debe redistribuir con el nodo C si la politica que se usa es la de derecha
4. Todas las anteriores son correctas
5. Algunas de las anteriores son correctas
6. **ninguna de las anteriores es correcta**

9. Un arbol b\* es más eficiente que un arbol b

1. porque tiene un algoritmo de inserción más eficiente
2. porque tiene un algoritmo de borrado más eficiente
3. porque tiene un algoritmo de busqueda más eficiente
4. no, un arbol b\* no es más eficiente que un arbol b
5. por todas las anteriores
6. por algunas de las anteriores
7. **por ninguna de las anteriores**

10. Un arbol b\*

1. puede ser un arbol binario de búsqueda
2. puede implementarse con una estructura de datos lineal
3. puede tener elementos repetidos, o sea implantar un indice secundario
4. puede tener acceso secuencia eficiente y rápido
5. Todas las anteriores son correctas
6. Algunas de las anteriores son correctas
7. **ninguna de las anteriores son correctas**

11. Suponga que el nodo terminal de un árbol B+ de prefijos simples de orden 7 tiene las claves enteras siguientes 10342, 10563, 11763, 12345, 13423, 13443 y se inserta en ese nodo la clave 15000

1. se produce overflow, y luego de dividir, al padre del nodo se sube la clave 12
2. se produce overflow, y luego de dividir, al padre del nodo se sube la clave 11
3. **se produce overflow, y luego de dividir se sube la clave 12345**
4. Todas las anteriores pueden ser correctas, depende el algoritmo utilizado
5. a y b pueden se correctas, dependen del algoritmo utilizado
6. a y c pueden ser correctas, dependen del algoritmo utilizado
7. b y c pueden ser correctas, dependen del algoritmo utilizado
8. ninguna de las anteriores es correcta

12. Cuando una clave “x” y otra clave “y” generan, por función de dispersión, diferente dirección, entonces:

1. Una de las dos no será almacenada en el archivo.
2. **Se puede producir un desborde.**
3. Se produce una colisión.
4. Hay más de una opción correcta.
5. Ninguna es correcta

13. El método de doble dispersión

1. Utiliza una segunda función de hash para ubicar a todos los registros del archivo
2. Utiliza una segunda función de hash para ubicar algunos registros del archivo
3. Evita generar zonas contiguas de nodos en overflow
4. todas las anteriores son validas
5. Las opciones a y b son verdaderas
6. Las opciones a y c son verdaderas
7. **Las opciones b y c son verdaderas**
8. Ninguna de las anteriores

14. En árboles B, la política izquierda o derecha, determina lo siguiente:

1. **Se intenta redistribuir con el hermano adyacente izquierdo, si no es posible, se intenta con el hermano adyacente derecho, si tampoco es posible, se fusiona con hermano adyacente izquierdo.**
2. Se intenta redistribuir con el hermano adyacente derecho, si no es posible, se intenta con el hermano adyacente izquierdo, si tampoco es posible, se fusiona con hermano adyacente derecho.
3. Se intenta fusionar con el hermano adyacente derecho, si no es posible, se intenta con el hermano adyacente izquierdo, si tampoco es posible, se redistribuye con hermano adyacente derecho.
4. Se intenta fusionar con el hermano adyacente izquierdo, si no es posible, se intenta con el hermano adyacente derecho, si tampoco es posible, se redistribuye con hermano adyacente izquierdo.

15. El hash con espacio de direccionamiento dinámico

1. **Encuentra los registros con un solo acceso a disco**
2. Se puede aplicar a cualquier clave
3. No requiere de espacio adicional
4. Todas las anteriores
5. Algunas de las anteriores son validas
6. Ninguna de las anteriores es valida

**2022**

12/05/2022

1. Un archivo denominado serie, tiene

1. **Acceso secuencial**
2. Acceso secuencial indizado
3. Acceso directo
4. Más de una respuesta es correcta
5. Ninguna respuesta es correcta

2. El proceso de merge

1. Debe realizarse sobre archivo ordenado físicamente por algún criterio
2. Puede realizarse sobre archivos ordenado por algún criterio
3. Necesita que los archivos sobre el cual se aplica este ordenado físicamente por algún criterio.
4. Todas las anteriores
5. Algunas de las anteriores
6. **Ninguna de las anteriores**

3. El proceso de baja lógica

1. Recupera el espacio del disco automáticamente
2. Es más eficiente que el proceso de baja física
3. Solo se aplica con registros de longitud fija
4. Solo se aplica con registros de longitud variable
5. Más de una es correcta
6. **Ninguna es correcta**

4. La técnica de recuperación de espacio denominada mejor ajuste

1. Permite borrar un registro de longitud fija
2. Permite borrar un registro de longitud variable
3. Permite recuperar espacio con un proceso de alta si los registros son de longitud fija
4. Todas las anteriores
5. Algunas de las anteriores
6. Ninguna de las anteriores

5. Un índice

1. Permite recorrer de forma más eficiente un archivo serie
2. Permite recorrer de forma más eficiente un archivo directo
3. Solo se aplica en archivos con registro de longitud variable
4. Todas las anteriores
5. Algunas de las anteriores
6. Ninguna de las anteriores

14/07/2022

1. Un archivo serie

1. **Tiene acceso secuencial y orden de búsqueda lineal**
2. Tiene acceso secuencial indizado y orden de búsqueda logarítmico
3. Tiene acceso secuencial y orden de búsqueda constante
4. Tiene acceso secuencial indizado y orden de búsqueda lineal
5. Todas las anteriores
6. Algunas de las anteriores
7. Ninguna de las anteriores

2. El proceso de actualización de un archivo con n archivos

1. Se puede aplicar con todos los archivos ordenados por algún criterio
2. Se puede aplicar con todos los archivos desordenados
3. Se puede aplicar con algunos ded los archivos ordenados por algún criterio y el resto desordenados
4. Se puede aplicar con cada uno de los n archivos ordenados por diferentes criterios
5. **Todas las anteriores son válidas**
6. algunas de las anteriores son válidas
7. Ninguna de las anteriores son válidas

3. Sea un archivo desordenado que contiene 2.000.000 de registros. En cada buffer de memoria asociado para acceso a disco caben 1000 registros, entonces, para encontrar un registro en particular, en el peor de los casos se hacen

1. **2.000.000 de lecturas y 2.000 accesos a disco**
2. 2.000.000 de lecturas y 2.000.000 de accesos a disco
3. 2.000 lecturas y 2.000 accesos a disco
4. No tengo datos suficientes para responder la pregunta
5. Hay datos suficientes, pero ninguna es correcta

4. Si se baja a un registro de un archivo secuencial indizado

1. Es igual de eficiente hacer una baja lógica que una baja física
2. Es más eficiente hacer una baja lógica que una baja física
3. Es menos eficiente hacer una baja lógica que una baja física
4. **Ninguna de las anteriores**

5. Si se realiza una baja física del décimo registro de un archivo que tiene 40.000 registros

1. Es posible recuperar el espacio que el registro utilizaba
2. Es posible que el archivo se desordene si estaba ordenado
3. Es posible que la baja sea más ineficiente que darlo de baja lógicamente
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. **Ninguna de las anteriores son válidas**

6. Si se realiza una baja lógica de un registro de un archivo

1. Se genera fragmentación interna
2. **Se genera fragmentación externa**
3. No se genera fragmentación
4. No tengo datos suficientes para responder

7. Un archivo con registros de longitud variable

1. Ocupa más espacio que el mismo archivo con registros de longitud fija
2. Ocupa el mismo espacio que el mismo archivo con registros de longitud fija
3. Ocupa menos espacio que el mismo archivo con registros de longitud fija
4. **No tengo datos suficientes para contestar**

8. Para poder realizar un algoritmo de corte de control sobre un archivo

1. Es suficiente que el archivo esté ordenado por al menos un criterio
2. Es necesario que el archivo esté ordenado por al menos un criterio
3. Es suficiente que el archivo esté ordenado por al menos dos criterios
4. Es necesario que el archivo esté ordenado por al menos dos criterios
5. Todas las anteriores son válidas
6. **Algunas de las anteriores son válidas**
7. Ninguna de las respuestas anteriores

9. Una clave secundaria

1. Identifica unívocamente un elemento del archivo
2. Identifica unívocamente varios elementos de un archivo
3. **Es un atributo del archivo donde está definida**
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. Ninguna de las respuestas anteriores vale

10. El proceso de modificación de un registro en un archivo

1. Puede modificar la clave primaria
2. Puede modificar una clave unívoca
3. Puede modificar una clave secundaria
4. Puede no modificar ninguna clave
5. Todas las anteriores son válidas
6. Algunas de las anteriores son válidas
7. Ninguna de las respuestas anteriores vale

11. Un índice

1. Se implementa con un vector
2. Se implementa con un árbol binario
3. Se implementa con un árbol b
4. Se implementa con un arbol b+
5. Todas las anteriores son válidas
6. Algunas de las anteriores son válidas
7. **Ninguna de las respuestas anteriores vale**

12. Un árbol binario paginado

1. Puede desbalancearse
2. Puede estar balanceado
3. Puede no estar balanceado
4. **Todas las anteriores son válidas**
5. Algunas de las anteriores son válidas
6. Ninguna de las respuestas anteriores vale

13. Dado un índice de un archivo. Para un momento determinado:

1. Si se implementó con árbol b tiene mayor altura que si se implementó con un árbol B+
2. Si se implementó con un árbol b tiene igual altura que si se implementó con un árbol B+
3. Si se implementó con un árbol b tiene menor altura que si se implemento con un árbol B+
4. **No tengo datos suficientes para responder**

14. Un árbol multicamino

1. Es un árbol binario
2. Es un árbol AVL
3. Es un árbol B+
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. **Ninguna de las anteriores son válidas**

15. Sea un árbol b de orden 50, si el nodo 15 tiene 49 claves y se inserta un nuevo registro en dicho nodo

1. **Se produce overflow y el nodo se divide de un nodo a dos nodos**
2. Se produce overflow y el nodo se redistribuye
3. Se produce overflow y se dividen de dos nodos a tres nodos
4. Ninguna es correcta

16. Sea un árbol B\* de orden 5. Sean A y B dos nodos adyacentes hermanos. A contiene las claves (1, 4, 6, 7), B contiene (15, 19, 26). En el padre de ambos nodos el separador es 10. Se llega la clave 8

1. El nodo A se divide y no se afecta al nodo B
2. **El nodo A redistribuye elementos con el nodo B y queda A= (1, 4, 6, 7) B= (10, 15, 19, 26) siendo 8 el promocionado al padre**
3. El nodo A queda con (1, 4, 6, 7, 8)
4. Ninguna respuesta es correcta

17. Sea un índice correspondiente a una clave candidata implementado sobre un atributo de tipo de dato integer.

1. Se puede implementar con un árbol b
2. Se puede implementar con un árbol b\*
3. Se puede implementar con un árbol b+
4. Se puede implementar con un árbol B+ de prefijos simples
5. Todas las anteriores
6. **Algunas de las anteriores**
7. Ninguna de las anteriores

18. La política de izquierda y derecha en la implementación de un árbol b\*

1. Genera árboles potencialmente más bajos en altura que otras políticas
2. Genera árboles con nodos con menos fragmentación externa que otras políticas
3. Es siempre más ineficiente algorítmicamente que otras políticas
4. Todas las anteriores
5. Algunas de las anteriores
6. Ninguna de las anteriores

19. La eficiencia de búsqueda en un árbol b\* con política de izquierda y derecha

1. Es de orden lineal
2. **Es de orden logarítmico**
3. Es de orden constante
4. No tengo datos suficientes para responder debido a la política utilizada

20. Suponga que sobre un nodo de un árbol B\* se produce overflow. En dicho caso se puede

1. Aplicar saturación progresiva encadenada
2. Aplicar doble dispersión
3. Aplicar un área de desborde separada para el nodo
4. Todas las anteriores
5. **Ninguna de las anteriores**

21. El acceso directo a un elemento de datos es

1. Más eficiente que el acceso secuencial
2. Más eficiente que el acceso secuencial indizado
3. Más eficiente que utilizar un árbol B+´
4. **Todas las anteriores son válidas**
5. Algunas de las anteriores son válidas
6. Ninguna de las anteriores son válidas

22. Una función de hash en teoría debe ser

1. **Uniforme y aleatoria**
2. Uniforme pero no aleatoria
3. Aleatoria pero no uniforme
4. No debe ser ni uniforme ni aleatoria

23. La densidad de empaquetamiento

1. Mide la relación entre espacio disponible y espacio utilizado
2. Mide la relación entre la cantidad de registros que tiene un archivo y la capacidad de almacenamiento del mismo
3. Es el cociente entre el número de registros del archivo disponible y el espacio disponible
4. **Todas las anteriores son válidas**
5. Algunas de las anteriores son válidas
6. Ninguna de las anteriores es válida

24. En caso de que se produzca overflow cuando se inserta sobre un nodo cualquiera

1. Se puede utilizar saturación progresiva para su tratamiento
2. Se puede utilizar saturación progresiva encadenada para su tratamiento
3. Se puede utilizar doble paginación para su tratamiento
4. Se puede utilizar área de desborde por separado para su tratamiento
5. **Todas las anteriores son válidas**
6. Algunas de las anteriores son válidas
7. Ninguna de las anteriores son válidas

25. El método de área de desborde por separado es

1. Más eficiente que el método de saturación progresiva
2. Más eficiente que el método de saturación progresiva encadenada
3. Más eficiente que el método de doble dispersión por separado
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. **Ninguna de las anteriores son válidas**

26. Cuando se produce una inserción en hash estático, sobre el archivo de datos

1. Se realiza una operación de escritura y una de lectura
2. **Se realiza al menos una operación de lectura y al menos una de escritura**
3. Se realiza una operación de lectura y al menos una de escritura
4. Se realiza una operación de escritura y al menos una de lectura

27. En hashing extensible, cuando la Densidad de empaquetamiento tiende a 1:

1. Se debe pasar por hash estático a dinámico
2. Se debe reformular el tamaño del nodo
3. Se debe aumentar la cantidad de registros por nodo
4. Algunas de las anteriores
5. **Ninguna de las anteriores**

28. El método de dispersión con espacio de direccionamiento dinámico, denominada has extensible:

1. Guarda los registros de forma ordenada por algún criterio
2. **Siempre requiere el uso de una estructura auxiliar**
3. Siempre requiere variar el tamaño del espacio de direcciones
4. Necesita de dos funciones de dispersión
5. Ninguna de las opciones anteriores es correcta

29. Con 10000 direcciones con capacidad para 4 registros cada una y 30000 claves para dispersar, entonces

1. La densidad de empaquetamiento es inferior al 50% pero mayor o igual al 25%
2. La densidad de empaquetamiento es inferior al 75% pero mayor o igual al 50%
3. La densidad de empaquetamiento es mayor que uno
4. **La densidad de empaquetamiento es superior o igual al 75%**
5. No se puede calcular

30. De los cuatro métodos de tratamiento de desbordes en hash estático vistos, cual de ellos potencialmente no genera zonas de overflow contiguas dentro del espacio de direccionamiento de la función de hash.

1. Saturación progresiva
2. Saturación progresiva encadenada
3. **Doble dispersión**
4. Área de desborde separado

**2023**

03/2023 – 06/07/2023

1. Dado un archivo maestro que almacena código de producto y stock, y dado dos archivos de detalle que almacenan fecha de venta, código de producto, y cantidad vendida. El archivo maestro ordenado por stock y los detalles por código de producto, entonces en la implementación de un algoritmo de actualización maestro/detalles:

1. Se pueden procesar ambos archivos de detalle simultáneamente.
2. Puede no ser necesario procesar todos los registros del archivo maestro.
3. Se deben procesar todos los registros de los archivos de detalle.
4. Hay más de una opción válida.
5. **Todas las opciones anteriores son correctas.**
6. Ninguna de las opciones es correcta.

2. La siguiente secuencia de instrucciones en un archivo con la siguiente estructura: código de producto y precio: assign(arch, “archivo.dat”); rewrite(arch); producto.codigo:= 1; producto.precio:= 5000; write(arch, producto); close(arch); reset(arch); writeln(filesize(arch));

1. Pierde información.
2. Modifica el código y el precio del último registro existente en el archivo.
3. Imprime 0 en pantalla.
4. Imprime 1 en pantalla.
5. No realiza cambio alguno sobre el archivo.
6. **Hay más de una opción válida.**
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

3. Dado un archivo organizado con una lista invertida interna al mismo archivo para recuperación de espacio libre:

1. El registro ubicado en la posición 1 se considera un registro especial que se utiliza como inicio de la lista.
2. Ante una nueva alta, se recorre el archivo de forma secuencial buscando un espacio libre.
3. Ante una nueva alta, si la lista interna de espacio libre está vacía se recorre el archivo secuencialmente hasta el final y ahí se agrega el nuevo registro.
4. Ante una baja, no es necesario actualizar la lista interna de espacio libre.
5. Ante una baja, el registro a borrar se marca para indicar que se encuentra eliminado del archivo.
6. Si se eliminan todos los registros del archivo, entonces se debe eliminar el registro en la posición 0 del archivo.
7. La lista invertida interna no puede estar vacía.
8. Hay más de una opción válida.
9. Todas las opciones anteriores son correctas.
10. **Ninguna de las opciones anteriores es correcta.**

4. Un índice primario es:

1. Una estructura de datos adicional que permite asociar una o varias claves primarias con una clave secundaria.
2. Una estructura de datos adicional que contiene el mismo volumen de información que el archivo original.
3. Una estructura de datos adicional que permite asociar una o varias claves secundarias con una clave primaria.
4. **Una estructura de datos adicional que establece un orden lógico del archivo original.**
5. Una estructura de datos adicional que permite relacionar una clave primaria con otra clave candidata.
6. Hay más de una opción válida.
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

5. Un árbol AVL:

1. Es un árbol binario en donde cada nodo puede tener más de un nodo padre.
2. **Es un árbol binario balanceado en altura.**
3. Es una estructura de datos lineal que se puede desbalancear.
4. Puede ser árbol B balanceado en altura.
5. Puede ser árbol B+ balanceado en altura.
6. Hay más de una opción válida.
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

6. Cuando se realiza un alta en un árbol B+ de prefijos simples:

1. No necesita la creación de más de un nodo.
2. Siempre se accede al nivel hoja.
3. Puede no ser necesario realizar una actualización en un nodo interno.
4. Nunca puede aumentar el nivel del árbol.
5. Siempre se promociona una copia de la expresión mínima de la clave.
6. **Hay más de una opción válida.**
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

7. Cuando se realiza una baja en un árbol B:

1. El elemento que se elimina no puede estar en un nodo interno del árbol.
2. **Puede existir underflow y propagarse hasta el nodo raíz.**
3. Siempre es necesario liberar un nodo.
4. El árbol desciende de nivel.
5. No es necesario acceder al nivel hoja.
6. Hay más de una opción válida.
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

8. Cuando se realiza un alta y produce una colisión en dispersión estática:

1. Siempre se genera desborde.
2. Nunca se genera un desborde.
3. **Se puede generar desborde.**
4. Se puede generar una colisión.
5. Hay más de una opción válida.
6. Todas las opciones anteriores son correctas.
7. Ninguna de las opciones anteriores es correcta.

9. Cuando una clave “x” y otra clave “y” generan, por Función de Dispersión, distinta dirección, entonces:

1. **Puede ocurrir un desborde.**
2. Puede ocurrir una colisión.
3. “x” e “y” no son claves sinónimas.
4. Todas las opciones anteriores son correctas.
5. Hay más de una opción válida.
6. Ninguna de las opciones anteriores es correcta.

10. Cuando se produce un desborde en dispersión dinámica:

1. En algunos casos aumenta la cantidad de bits asociado al bloque en donde sucede el desborde.
2. Siempre se duplica la tabla asociada al archivo de datos.
3. Se debe generar un nuevo bloque para el archivo de datos.
4. Puede no ser necesario duplicar la tabla asociada al archivo de datos.
5. Puede no ser necesario escribir en el archivo de datos.
6. Hay más de una opción válida.
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

07/09/2023

1. Dado el siguiente árbol B+ de orden 5, la operación debaja del valor 52:

Nodo 0: 4, i, 1 (23) 2 (52) 3 (73) 4 (88) 5

Nodo 1: 4, h, (5)(9)(15)(20) -> 2

Nodo 2: 3, h, (23)(32)(41) -> 3

Nodo 3: 2, h, (53)(62) -> 4

Nodo 4: 1, h, (78) -> 5

Nodo 5: 1, h, (90) -> -1

1. Genera la lectura del nodo 0 y la estructura del nodo 0.
2. Genera la lectura del nodo 0 y la lectura del nodo 3.
3. No genera lecturas y escrituras.
4. Genera la lectura del nodo 0, lectura del nodo 3 y no genera escrituras.
5. Genera la lectura del nodo 0, lectura del nodo 2 y no genera escrituras.
6. Hay más de una opción válida.
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

2. Dado un archivo organizado con una lista invertida interna al mismo archivo para recuperación de espacio libre:

1. Ante una nueva alta, se recorre el archivo de forma secuencial buscando un espacio libre.
2. Ante una nueva alta, se modifica el registro de la posición 0.
3. Ante una baja de un elemento existente en el archivo, puede ser que no se modifique el registro de la posición 0.
4. Si se eliminan todos los registros del archivo, la operación de filesize imprime 0.
5. Hay más de una opción válida.
6. Todas las opciones anteriores son correctas.
7. **Ninguna de las opciones anteriores es correcta.**

3. Según el estado de un archivo directo para dispersión extensible y bloques o cubetas con capacidad para 3 registros:

1. El alta de la clave 30845 (…1) no produce overflow.
2. **El alta de la clave 12640 (…0) no produce overflow.**
3. El alta de la clave 30845 (…1) produce overflow, pero no es necesario duplicar la tabla.
4. El alta de la clave 12640 (…0) produce overflow, pero no es necesario duplicar la tabla.
5. Hay más de una opción válida.
6. Todas las opciones anteriores son correctas.
7. Ninguna de las opciones anteriores es correcta.

4. Un árbol AVL es un (utilizar sólo 5 palabras, no más):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Un índice secundario es:

1. Una estructura de datos adicional que permite asociar una o varias claves primarias con una clave candidata.
2. Una estructura de datos adicional que contiene el mismo volumen de información, que el archivo original.
3. Una estructura de datos adicional que permite asociar una o varias claves secundarias con una clave primaria.
4. Hay más de una opción válida.
5. Todas las opciones anteriores son correctas.
6. **Ninguna de las opciones anteriores es correcta.**

6. Cuando se busca un elemento en un árbol B+:

1. Puede necesitar la creación de un nodo.
2. Puede no ser necesario acceder al nivel hoja.
3. Puede ser necesario realizar una actualización en un nodo interno.
4. Hay más de una opción válida.
5. Todas las opciones anteriores son correctas.
6. **Ninguna de las opciones anteriores es correcta.**

**Simulacro**

1. Un archivo directo

1. Tiene acceso Directo y orden de búsqueda lineal
2. Tiene acceso Directo y orden de búsqueda logarítimico
3. Tiene acceso Directo y orden de búsqueda exponencial
4. Tiene acceso secuencial indizado y orden de búsqueda lineal
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

2. Un proceso de merge de n archivos, para que sea lo más eficiente posible en términos de rendimiento/performance

1. Se debe aplicar con todos los archivos ordenados por algún criterio
2. Se debe aplicar con todos los archivos desordenados
3. Se debe aplicar con algunos de los archivos ordenados por algún criterio y el resto desordenados
4. Se debe aplicar con cada uno de los n archivos ordenados por diferentes criterios
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

3. Si se realiza la baja de un registro en un archivo directo

1. Es igual de eficiente hacer una baja lógica que una baja física
2. Es más eficiente hacer una baja lógica que una baja física considerando el espacio ocupado
3. Es menos eficiente hacer una baja lógica que una baja física considerando el rendimiento/performance
4. **Ninguna de las anteriores**

4. Un archivo directo con registros de longitud variable

1. Puede ocupar más espacio que el mismo archivo con registros de longitud fija
2. Puede ocupar el mismo espacio que el mismo archivo con registros de longitud fija
3. Ocupa menos espacio que el mismo archivo con registros de longitud fija
4. **Ninguna de las anteriores**

5. Para poder realizar un algoritmo de bajas sobre un archivo

1. Es suficiente que el archivo este ordenado por al menos un criterio
2. Es necesario que el archivo esté ordenado por al menos un criterio
3. Es suficiente que el archivo esté ordenado por al menos dos criterios
4. Es necesario que el archivo esté ordenado por al menos dos criterios
5. Todas las anteriores son válidas
6. **Algunas de las anteriores son válidas**
7. Ninguna de las respuestas anteriores

6. Una clave

1. Identifica unívocamente un elemento del archivo
2. Identifica varios elementos de un archivo
3. Está constituida por al menos dos atributos del archivo donde está definida
4. Todas las anteriores
5. **Algunas de las anteriores**
6. Ninguna de las anteriores

7. Un índice secundario implementado con un árbol binario

1. Puede desbalancearse
2. Puede estar balanceado
3. Puede balancearse
4. Se desbalancea fácilmente
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

8. Un árbol

1. Puede desbalancearse
2. Puede estar balanceado
3. Está balanceado en altura
4. Puede no estar balanceado
5. Todas las anteriores
6. **Algunas de las anteriores**
7. Ninguna de las anteriores

9. Dado un índice de un archivo con registros de longitud variable

1. Sus registros pueden ser de longitud variable
2. Sus registros deben ser de longitud variable
3. Sus registros pueden ser de longitud fija
4. **Sus registros son de longitud fija**

10. Un árbol

1. **Puede ser árbol binario y AVL**
2. Puede ser B\* y B+
3. Puede ser B y B+
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. Ninguna de las respuestas anteriores vale

11. La saturación progresiva encadenada

1. Se usa en caso de colisión
2. Se usa en caso de colisión y saturación
3. Se usa en caso en caso de saturación
4. **Se puede usar en caso de saturación**
5. Todas las anteriores
6. Algunas de las anteriores
7. Ninguna de las anteriores

12. Una función de hash uniforme y aleatoria

1. **Tiene un promedio de 1, de acceso a registros**
2. Tiene un promedio 1 y 2, de acceso a registros
3. Tiene un promedio menor a 1, de acceso a registros
4. Ninguna de las anteriores

13. La densidad de empaquetamiento

1. Se modifica con cada alta o baja, utilizando hashing extensible
2. Se modifica con cada alta, utilizando hashing extensible
3. Se modifica con cada baja, utilizando hashing extensible
4. No se modifica con altas o bajas, utilizando hashing extensible
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**

14. Si hubo colisión con hashing dinámico:

1. Ocurre overflow
2. No ocurre overflow
3. Se trata con un algoritmo específico de tratamiento de colisiones
4. Algunas de las anteriores
5. **Ninguna de las anteriores**

**Sin Fecha**

1. Un índice secundario es:

1. **Una estructura de datos adicional que permite asociar una o varias claves primarias con una clave secundaria.**
2. Una estructura de datos adicional que contiene el mismo volumen de información que el archivo original.
3. Una estructura de datos adicional que permite asociar una o varias claves secundarias con una clave primaria.
4. Una estructura de datos adicional que establece un orden lógico del archivo original.
5. Una estructura de datos adicional que permite relacionar una clave primaria con otra clave candidata.
6. Hay más de una opción válida.
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

2. Cuando se realiza una baja en un árbol B+ de prefijos simples:

1. Necesita la creación de más de un nodo.
2. Siempre se accede al nivel hoja.
3. Es necesario realizar una actualización en un nodo interno.
4. Nunca puede aumentar el nivel del árbol.
5. Se promociona una copia de la expresión mínima de la clave.
6. Hay más de una opción válida.
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

3. Un árbol AVL:

1. Es un árbol que posee más de 2 descendientes.
2. Es un árbol no binario balanceado en altura.
3. Es una estructura de datos lineal que se puede desbalancear.
4. Puede ser árbol B balanceado en altura.
5. Puede ser árbol B no balanceado en altura.
6. Hay más de una opción válida.
7. Todas las opciones anteriores son correctas.
8. **Ninguna de las opciones anteriores es correcta.**

4. Cuando se realiza un alta en un árbol B+:

1. Puede requerir la liberación de un nodo.
2. Puede no ser necesario acceder al nivel hoja.
3. Siempre es necesario realizar una actualización en un nodo interno.
4. **Puede aumentar el nivel del árbol.**
5. Hay más de una opción válida.
6. Todas las opciones anteriores son correctas.
7. Ninguna de las opciones anteriores es correcta.

5. Cuando en un árbol B ocurre un desborde:

1. Se promociona la expresión mínima de la clave.
2. Se aplica redistribución de claves con el hermano derecho adyacente.
3. Puede no ser necesario crear un nuevo nodo.
4. **Siempre se debe generar un nuevo nodo.**
5. Hay más de una opción válida.
6. Todas las opciones anteriores son correctas.
7. Ninguna de las opciones anteriores es correcta.

6. La segunda función de dispersión en Doble Dispersión:

1. Se aplica a todas las claves a dispersar.
2. Retorna la dirección física en donde residirá la clave.
3. Retorna un valor al cual se le aplica la primera función de dispersión.
4. No existe una segunda función de dispersión en la técnica de Doble Dispersión.
5. Retorna un valor de desplazamiento que se debe utilizar para resolver un desborde.
6. Hay más de una opción válida.
7. Todas las opciones anteriores son correctas.
8. Ninguna de las opciones anteriores es correcta.

7. El proceso de baja lógica en un archivo de datos desordenado

1. Mantiene el archivo desordenado
2. Recupera el espacio para que otro registro pueda utilizarlo
3. En general, es el proceso más eficiente de baja en términos de tiempo
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. Ninguna de las anteriores son válidas

8. Si se utiliza el método de alta de registros con recuperación de espacio borrado

1. Se complementa con el proceso de baja lógica
2. Es más ineficiente en términos de tiempo de procesamiento que un proceso de alta que no recupera espacio
3. Puede tener variantes como primer ajuste, mejor ajuste y peor ajuste
4. **Todas las anteriores son válidas**
5. Algunas de las anteriores son válidas
6. Ninguna de las anteriores son válidas

8. La técnica de prefijos simples

1. Se aplica a todo tipo de árboles
2. Puede aplicarse para tratar colisiones y overflow en hash estático
3. Es una variante de arboles b\* para tener acceso secuencial a los datos
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. **Ninguna de las anteriores son válidas**

9. Los archivos serie

1. Tienen acceso secuencial
2. Pueden recuperar un registro en un solo acceso
3. Por concepción del archivo, no pueden aplicar búsquedas binarias
4. **Todas las anteriores son válidas**
5. Algunas de las anteriores son válidas
6. Ninguna de las anteriores son válidas

10. La densidad de empaquetamiento es un concepto que

1. Indica la relación entre espacio utilizado y disponible de un archivo que utilice hash estático
2. Es un parámetro de Hashing que indica la capacidad restante de un archivo.
3. Se modifica cada vez que se realiza una operación sobre el archivo.
4. **Todas las anteriores son válidas**
5. Algunas de las anteriores son válidas
6. Ninguna de las anteriores son válidas

11. El método de tratamiento de desborde de doble paginación

1. Es un método más eficiente que el de saturación progresiva encadenada
2. Es un método que evita la generación de zonas contiguas de overflow
3. Es un método que no utiliza una zona de overflow separada como el área de desborde
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. Ninguna de las anteriores son válidas

12. Una política de hash dinámica

1. Siempre encuentra los registros en un acceso a disco
2. Es más ineficiente para la inserción de registros
3. Cumple todas las propiedades definidas para hash
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. Ninguna de las anteriores son válidas

13. Un árbol binario paginado

1. Se encuentra siempre balanceado
2. Es un requerimiento que sea balanceado para ser un árbol binario de búsqueda
3. Permiten generar la estructura auxiliar para un archivo secuencial, y asi tener acceso indizado
4. Todas las anteriores son válidas
5. Algunas de las anteriores son válidas
6. Ninguna de las anteriores son válidas

14. Con respecto a un árbol B\*:

1. Es más eficiente realizar una búsqueda sobre un árbol B que sobre un árbol B\*.
2. **La altura puede ser inferior a la de un árbol B porque los elementos se distribuyen más eficientemente en los nodos.**
3. Permite acceder secuencialmente a los elementos del árbol.
4. Ninguna de las anteriores

15. Un archivo fragmentado

1. Debe compactarse para optimizar el espacio utilizado
2. No debe compactarse para optimizar el espacio utilizado
3. **A veces puede compactarse**
4. Nunca debe compactarse
5. Algunas de las anteriores

16. Un algoritmo de actualización maestro-detalles

1. Requiere que todos los archivos tengan la misma estructura
2. **Puede realizarse entre archivos con diferente estructura**
3. Requiere que los archivos estén ordenados
4. Requiere que los archivos estén desordenados
5. Algunas de las anteriores

17. Un archivo serie

1. Está ordenado
2. **Puede ordenarse**
3. Requiere ordenarse
4. No requiere ordenarse
5. No está ordenado
6. Algunas de las anteriores

18. Dado un archivo con 1000 registros

1. Siempre se puede llevar a memoria RAM para hacer búsquedas eficientes
2. **No siempre se puede llevar a memoria RAM para hacer búsquedas eficientes**
3. Siempre puede realizarse búsqueda dicotómica
4. No puede realizarse búsqueda dicotómica
5. Algunas de las anteriores

**Más choices**

1. Un algoritmo de búsqueda en un archivo:

1. Es más eficiente si el archivo está ordenado
2. **Puede ser más eficiente si se considera como precondición que el archivo está ordenado.**
3. Es igual de eficiente si el archivo está ordenado o desordenado.
4. Ninguna de las anteriores

2. Cuál de las siguientes definiciones corresponde a archivo:

1. Colección de registros que abarca un conjunto de entidades con ciertos aspectos en común y organizados para un propósito particular.
2. Colección de registros semejantes almacenados en disco rígido.
3. Colección de registros del mismo tiempo almacenados en un dispositivo de memoria secundaria.
4. **Todas las opciones son correctas**

3. El proceso de baja en un archivo con registros de longitud variable:

1. Puede recuperar el espacio disponible con nuevas altas
2. Puede recuperar el espacio disponible compactando periódicamente el archivo
3. Puede recuperar el espacio disponible compactando el archivo ante cada baja
4. **Todas las anteriores**

4. Una clave candidata:

1. Admite repeticiones de valores
2. Admite repeticiones de campos
3. **Podría haber sido elegida como clave primaria**
4. Tiene exactamente las mismas características que la clave primaria

5. El proceso de compactación de un archivo tiene sentido ser aplicado:

1. Luego de realizar una operación de alta
2. **Luego de realizar una operación de baja lógica**
3. Luego de realizar una operación de baja físicamente
4. Luego de realizar una operación de modificación

6. Un archivo físicamente ordenado

1. Es más fácil de recorrer
2. Permite búsqueda binaria
3. Permite búsqueda binaria solo si las altas mantienen el archivo ordenado
4. No permite búsqueda binaria si hay bajas lógicas
5. **Algunas de las anteriores**
6. Ninguna de las anteriores

7. La función EOF

1. Puede volver verdadero después de un reset
2. Puede devolver falso después de una lectura
3. Puede devolver verdadero después de una lectura
4. Devuelve verdadero si estás al final del archivo
5. Devuelve falso si no estás en el final del archivo
6. **Todas las anteriores**
7. Algunas de las anteriores
8. Ninguna de las anteriores

8. Es posible aplicar la búsqueda binaria en

1. Archivos desordenados con registros de longitud variable
2. Archivos desordenados con registros de longitud fija
3. Archivos ordenados con registros de longitud variable
4. **Archivos ordenados con registros de longitud fija**
5. Ninguna de las anteriores es válida

9. Un archivo organizado con registros de longitud variable

1. No permite realizar bajas lógicas
2. **Optimiza la utilización de espacio en disco**
3. No permite realizar bajas físicas
4. Solo acepta altas al final del archivo

10. La fragmentación en un archivo de longitud fija

1. Dificulta la baja física
2. Dificulta la baja lógica
3. **Produce pérdida de espacio**
4. Es menor utilizando primer ajuste
5. Algunas de las anteriores
6. Ninguna de las anteriores

11. Cuáles propiedades corresponder a un árbol B\* de orden M

1. La diferencia máxima de altura entre los dos subárboles cualesquiera que comparten raíz es 1
2. Un nodo terminal tiene como mínimo [M/2]-1 claves
3. Cada nodo puede tener como máximo M hijos
4. Un nodo no terminal que tiene K descendientes debe tener K-1 claves
5. Ninguna de las anteriores

12. Un árbol B

1. Puede ser un árbol AVL
2. Puede guardarse en memoria RAM
3. Puede implementar una clave no unívoca
4. Puede tener acceso secuencial eficiente y rápido
5. Todas las anteriores son correctas
6. **Algunas de las anteriores son correctas**
7. Ninguna de las anteriores son correctas

13. Un árbol que se encuentra balanceado

1. Puede ser un árbol binario
2. Puede ser un árbol multicamino
3. Puede ser un árbol B+
4. Puede ser un árbol B\*
5. **Todas las anteriores son correctas**
6. c y d son correctas
7. b, c y d son correctas
8. b y d son correctas
9. Ninguna de las anteriores es correcta

14. Un archivo tratado con hash estático que tiene una densidad del 10%

1. Tiene mucha fragmentación interna
2. Tiene mucha fragmentación externa
3. Presenta un nivel de colisiones bajo
4. Presenta un nivel de overflow bajo
5. Todas las anteriores
6. b y c son correctas
7. a y c son correctas
8. **a, c y d son correctas**

15. La función de hashing

1. Afecta a la densidad de empaquetamiento
2. Puede afectar a la densidad de empaquetamiento
3. Afecta la densidad de empaquetamiento sólo en casos especiales
4. Afecta la densidad de empaquetamiento en la mayoría de los casos
5. Todas las anteriores
6. Algunas de las anteriores
7. **Ninguna de las anteriores**